

### AMENDMENTS TO THE CLAIMS

Please add or amend the claims to read as follows, and cancel without prejudice or disclaimer to resubmission in a divisional or continuation application claims indicated as cancelled:

1-7. (Cancelled)

8. (Currently amended) A method of managing a plurality of queues of received packets, the method comprising:

sequentially inserting said packets into a circular first-in-first-out memory scheme; and

selectively associating said packets with said plurality of queues by updating queue information that links between a packet being inserted into said memory scheme and a previous packet associated with the same queue as the packet being inserted.

9. (Previously presented) The method of claim 8 comprising removing said packets from said memory scheme according to said plurality queues.

10. (Previously presented) The method of claim 8, wherein said plurality of queues comprises a plurality of independent first-in-first-out queues.

11. (Previously presented) The method of claim 8, wherein sequentially inserting said packets comprises:

inserting a start of a packet at a memory word indicated by a write pointer; and  
advancing said write pointer to indicate a memory word located after a memory word containing an end of said packet.

12. (Previously presented) The method of claim 11, wherein sequentially inserting said packets comprises advancing a delete pointer jointly with said write pointer such that an empty zone of a predetermined length is maintained between said write pointer and said delete pointer.

13. (Previously presented) The method of claim 12 comprising dropping an indicated packet corresponding to a memory word indicated by said delete pointer, if said indicated packet is linked to one of said plurality of queues.

14. **(Previously presented)** The method of claim 13 comprising determining if said indicated packet is linked to one of said plurality of queues by comparing a location of the memory word indicated by said delete pointer to a location indicated by a head pointer of a queue associated with said memory word.
15. **(Previously presented)** The method of claim 13 comprising determining a queue associated with said memory word based on information associating between a plurality of memory words and said plurality of queues.
16. **(Currently amended)** The method of claim 8, wherein ~~said associating comprises updating queue information associating between said packets and said plurality of queues~~ updating said queue information comprises updating a linking pointer associated with said previous packet to identify a location of the packet being inserted.
17. **(Currently amended)** The method of claim ~~16~~, wherein ~~updating said queue information comprises updating queue information corresponding to a packet being inserted into said memory scheme~~ comprising identifying said previous packet based on a tail pointer corresponding to the queue associated with the packet being inserted.
18. **(Currently amended)** The method of claim ~~16~~, wherein ~~updating said queue information comprises updating queue information corresponding to a packet being removed from said memory scheme~~ comprising managing a plurality of header pointers identifying the beginning of said plurality of queues, respectively.
19. **(Currently amended)** The method of claim ~~16~~, ~~wherein said queue information comprises~~ comprising updating one or more values corresponding to at least one of a header pointer of a queue, a tail pointer of said queue, and a link pointer linking between two consecutive packets of said queue.
20. **(Previously presented)** The method of claim 8, wherein a capacity of said memory scheme is predetermined based on an expected rate for receiving said packets into said memory scheme.
21. **(Currently amended)** A system for managing a plurality of queues of received packets, the system comprising:
  - a circular first-in-first-out memory scheme to sequentially store said packets;
  - and

a memory manager able to associate said packets with said plurality of queues by updating queue information that links between a packet being inserted into said memory scheme and a previous packet associated with the same queue as the packet being inserted.

22. **(Previously presented)** The system of claim 21, wherein said memory manager is able to remove said packets from said memory scheme according to said plurality of queues.
23. **(Previously presented)** The system of claim 21, wherein said plurality of queues comprises a plurality of first-in-first-out queues.
24. **(Previously presented)** The system of claim 21, wherein said memory scheme comprises a write pointer to indicate a memory word for inserting a word of a received packet.
25. **(Previously presented)** The system of claim 24, wherein said memory scheme comprises a delete pointer to be advanced jointly with said write pointer such that an empty zone of a predetermined length is maintained between said write pointer and said delete pointer.
26. **(Previously presented)** The system of claim 25, wherein said memory manager is able to drop an indicated packet corresponding to a memory word indicated by said delete pointer, if said indicated packet is linked to one of said plurality of queues.
27. **(Previously presented)** The system of claim 26, wherein said memory manager is able to determine if said indicated packet is linked to one of said plurality of queues by comparing a location of the memory word indicated by said delete pointer to a location indicated by a head pointer of a queue associated with said memory word.
28. **(Previously presented)** The system of claim 26, wherein said memory manager is able to determine a queue associated with said memory word based on information associating between a plurality of memory words and said plurality of queues.
29. **(Currently amended)** The system of claim 21, wherein said memory manager is able to ~~update queue information associating between said packets and said plurality of queues~~ update a linking pointer associated with said previous packet to identify a location of the packet being inserted.

30. **(Currently amended)** The system of claim ~~29~~21, wherein said memory manager is able to ~~update queue information corresponding to a packet being inserted into said memory scheme~~ manage a plurality of header pointers identifying the beginning of said plurality of queues, respectively.
31. **(Currently amended)** The system of claim ~~29~~21, wherein said memory manager is able to ~~update queue information corresponding to a packet being removed from said memory scheme~~ identify said previous packet based on a tail pointer corresponding to the queue associated with the packet being inserted.
32. **(Currently amended)** The system of claim 21, wherein said ~~queue information~~ comprises memory manager is able to maintain one or more values corresponding to at least one of a header pointer of a queue, a tail pointer of said queue, and a link pointer linking between two consecutive packets of said queue.
33. **(Previously presented)** The system of claim 21, wherein a capacity of said memory scheme is predetermined based on an expected rate for receiving said packets into said memory scheme.
34. **(New)** The method of claim 9, wherein said removing comprises removing a packet identified by a header pointer of a queue, and updating said header pointer to identify a successive packet which is linked to the packet being removed.
35. **(New)** The method of claim 13, wherein dropping said indicated packet comprises updating a header pointer of a queue associated with said indicated packet to identify a successive packet, which is linked to said indicated packet.
36. **(New)** The method of claim 13, wherein dropping said indicated packet comprises dropping an oldest packet stored in said memory.
37. **(New)** The system of claim 22, wherein said memory manager is able to remove a packet identified by a header pointer of a queue, and update said header pointer to identify a successive packet which is linked to the packet being removed.
38. **(New)** The system of claim 26, wherein said memory manager is able to update a header pointer of a queue associated with said indicated packet to identify a successive packet, which is linked to said indicated packet.
39. **(New)** The system of claim 26, wherein said indicated packet comprises an oldest packet stored in said memory.